The present invention is not anticipated by, nor obvious in view of, the references relied upon in the Office Action, as this prior art references do not disclose or suggest the claimed features of the present invention.

The Applicant respectfully submits that claims 3 and 12 are not indefinite. In particular, the phrases "complex folders" and "in-line views" are terms that are well-known to one of skill in the art. An example of the well-known usage of the phrase "in-line views" is shown in APPENDIX A, which is an extract from the Oracle 8I SQL Reference manual, published September 2000. An example of the well-known usage of the phrase "complex folders" is shown in APPENDIX B, which is an extract from the Oracle9i Discoverer Administrator Administration Guide. The examples clearly show the well-known meaning of each phrase to one of skill in the art.

The Applicant respectfully submits that the present invention according to claims 1-15 is not anticipated by Dalal. Dalal discloses a method and system for efficiently performing database table aggregation. In a preferred embodiment, an aggregation facility efficiently aggregates a source table using indices on an aggregated column of the source table and a grouping column of the source table. The facility uses the index on the aggregated column to identify the contents of the aggregated column in each row of the source table. The facility further uses information derived from the index on the grouping column to identify the contents of the grouping column in each row of the source table. For each row of

the source table, the facility aggregates the identified aggregated column contents into a result value for the identified grouping column contents. In a further preferred embodiment, the facility generates a relation mapping from source table row to grouping column, which the facility uses to identify the contents of the grouping column in each row of the source table. In a further preferred embodiment, the facility may be used to perform multiple-level aggregations, as well as aggregations in which there are multiple grouping columns, multiple aggregated columns, and/or multiple result columns.

Dalal further discloses a query processor that manipulates data stored in one or more tables in response to instructions, called "queries", from users.

By contrast, the present invention, for example, according to claim 1, requires a query generator for generating a query for obtaining selected data from a database, the database having a number of tables in which data is stored, the query generator comprising a processor which is coupled to the database in use, the processor being adapted to: receive an input indicating the selected data to be obtained, analyze the input and determine whether the input requires a joining of data in different tables, and an aggregation step, and, if so, causing the processor to generate a query, the query being adapted to cause the database to: aggregate the data within each of the tables as required, and join the aggregated data, the joined aggregated data representing the selected data.

Dalal does not disclose a query generator for generating a query. Rather Dalal discloses a query processor that responds to queries that are provided to it.

Dalal does not disclose determining whether input requires a joining of data in different tables. Dalal only discloses the aggregation of data in a single table, not the aggregation of data in multiple tables, which requires the joining of data in different tables.

Dalal does not disclose generating a query. Rather Dalal discloses a query processor that responds to queries that are provided to it.

Dalal does not disclose aggregating the data within multiple tables and joining the aggregated data. Dalal only discloses the aggregation of data in a single table.

Similarly, claim 9 recites a query generator for generating a structured query for obtaining selected data from the database, the query generator comprising a processor adapted to: receive an input representing the query to be generated, analyze the input and determine whether the input requires a joining of data in different tables, and an aggregation step, and if so, causing the processor to generate a structured query, wherein the database processor responds to the structured query to: aggregate the data within each of the tables as required, and join the aggregated data, the joined aggregated data representing the selected data.

Dalal does not disclose a query generator for generating a structured query. Rather Dalal discloses a query processor that responds to queries that are provided to it.

Dalal does not disclose determining whether input requires a joining of data in different tables. Dalal only discloses the aggregation of data in a single table, not the aggregation of data in multiple tables, which requires the joining of data in different tables.

Dalal does not disclose generating a structured query. Rather Dalal discloses a query processor that responds to queries that are provided to it.

Dalal does not disclose aggregating the data within multiple tables and joining the aggregated data. Dalal only discloses the aggregation of data in a single table.

Thus, the present invention, according to claims 1 and 9, is not anticipated by Dalal. Likewise, the present invention, according to claims 2-8 and 1-15, which depend from claims 1 or 9, is not anticipated by Dalal.

In view of the above, it is respectfully submitted that the present invention is allowable over the references relied upon in the Office Action. Accordingly, favorable reconsideration of this case and early issuance of the Notice of Allowance are respectfully requested.
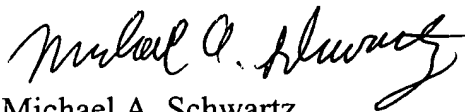
Additional Fees:

The Commissioner is hereby authorized to charge any insufficient fees or credit any overpayment associated with this application to Deposit Account No. 19-5127 (19111.0045).

Conclusion

In view of the foregoing, all of the Examiner's rejections to the claims are believed to be overcome. The Applicants respectfully request reconsideration and issuance of a Notice of Allowance for all the claims remaining in the application. Should the Examiner feel further communication would facilitate prosecution, he is urged to call the undersigned at the phone number provided below.

Respectfully Submitted,

Michael A. Schwartz
Reg. No. 40,161

Dated: December 20, 2002

Swidler Berlin Shereff Friedman, LLP
3000 K Street, N.W., Suite 300
Washington, D.C. 20007
(202) 424-7500

Version With Markings To Show Changes Made

4. (Amended) A query generator according to claim 1, wherein the aggregation comprises at least one of: generating an average of the data, generating a sum of the data, generating a standard deviation of the data, and generating an evaluation of either a maximum or a minimum of the data.

6. (Amended) A query generator according to claim 5, wherein the input is generated by selecting items from a list of possible items, each item representing data contained within a respective database table, [and/or] an action to be performed on data within the database, or both data contained within a respective database table and an action to be performed on data within the database.

7. (Amended) [Apparatus] A query generator according to claim 1, wherein the query being generated as an SQL query.

8. (Amended) [Apparatus] A query generator according to claim 1, wherein the aggregation step requires the aggregation of data in different tables.

10. (Amended) A system according to claim 9, wherein the database processor [comprises the processor.] and the processor of the query generator are a same processor.

13. (Amended) A system according to claim 9, wherein the aggregation comprises at least one of: generating an average of the data, generating a sum of the data, generating a standard deviation of the data, and generating a maximum and a minimum evaluation of the data.

14. (Amended) A system according to claim 9, the query generator further comprising an input device coupled to the processor of the query generator to allow a user to generate the input.

15. (Amended) A system according to claim 14, wherein the input is generated by selecting items from a list of possible items, each item representing data contained within a respective database table, [and/or] an action to be performed on data within the database, or both data contained within a respective database table and an action to be performed on data within the database.